uc3m | Universidad Carlos III de Madrid

Systems and Automation Department
Industrial Electronics and Automation Engineering
Universidad Carlos III de Madrid

Mobile Warning Application

Bachelor Thesis

Author: Irene Salazar Medina
Supervisor: Eng. Ahmed Hussein, M.Sc.
Submission Date: 31 July, 2017

uc3m | Universidad Carlos III de Madrid

Systems and Automation Department
Industrial Electronics and Automation Engineering
Universidad Carlos III de Madrid

Mobile Warning Application

Bachelor Thesis

Author: Irene Salazar Medina
Supervisor: Eng. Ahmed Hussein, M.Sc.
Submission Date: 31 July, 2017

This is to certify that:

(i) the thesis comprises only my original work toward the Bachelor Degree

(ii) due acknowledgment has been made in the text to all other material used

---

Irene Salazar Medina
31 July, 2017

Abstract

Pedestrians have been proved to get distracted on public ways, mostly by using smartphones, causing high danger when it comes to roads. In order to preserve their well-being, an application for Android has been developed to send a warning to the user when it detects a collision with a vehicle and the user is utilising and being distracted by the device. Considering the increasing presence of automated systems in vehicles, the application will also send the warning to the car in question for the same purpose, as this action may trigger a safety procedure in the vehicle that further increases road safety. The result is a free application that consumes few resources, employable on P2V communications and can be installed on most Android devices.

Resumen

Varios estudios demuestran que muchos peatones se distraen en las vías públicas, mayormente por el uso de teléfonos inteligentes, causando situaciones peligrosas en las calles donde circulan vehículos. Para asegurar su bienestar, se ha diseñado una aplicación de Android que manda notificaciones cuando el peatón se encuentra en una sitación de peligro con un vehículo y al mismo tiempo se encuentra utilizando el dispositivo. El resultado es una aplicación gratis que consume poca batería y se puede instalar en la mayoría de los dispositivos Android

Contents

Chapter 1

Introduction

1.1 Motivation

This project originally emerged from the growing concern for the safety of pedestrians' lives, due to the usage of smartphones in the street. These devices have been proven to distract people, making them unaware of their surroundings.

The presence of smartphones in people's daily lives has been on the rise, ever since these devices first appeared in the mid 90's. By the third quarter of 2012, over 1 billion units were being used worldwide [1]. In Spain, 71% of adults have reported they owned said devices [2].

This proves an issue in the public road environment, where people need to be aware of their surroundings and follow the rules of the road. Reports show that around 17% of monitored pedestrians had been using their smartphones while crossing the street, this percentage is increased to 22% for the younger population (ages 25-35) [3].

In addition to this, the aim of the project has expanded to include the appearance and increasing popularity of connected cars and autonomous vehicles. Studies show that connected car sales have the potential to build considerable revenue in the global car market, with an annual growth rate of up to 24.3% [4].

Meanwhile, the development of driverless vehicles has begun a competition between several car manufacturers. General Motors have invested more than 1500 million into building their first self-driving cars [5, 6, 7]. BMW announced that it will launch its driverless vehicle, the iNext, in 2021 [8]. Uber has even revealed its determination on providing a self-driving taxi network to ensure the survival of the company [9]. Most recently, Ford has disclosed that it will be mass-producing fully autonomous vehicles by 2021 [10].

In light of the arrival of these new vehicles with an internet connection, the project will be designed to exploit the opportunities that this technology provides. The perception systems of both driven and driverless cars require visibility, meaning that situations in

which the pedestrian is not detectable are perilous. This project attempts to implement a solution to this sensor limitation as well.

Therefore, to guarantee the safety of the pedestrian, the project will warn both the smartphone user and the car of the impending danger, in the case that a future collision is detected.

1.2 Objectives

This thesis builds on an application that is already in development, but still incomplete. Said application is capable of calculating the trajectories of the car and the pedestrian and recognising whether there is a probable collision. However, this is only done if the application is in the foreground, which is counter-productive since it forces the user to actively use an application that will warn them of collisions.

Therefore, the project's main goal is to implement a way for said calculations to be computed in the background, so the application does not have to be in the foreground, distracting the pedestrian from their surroundings.

Since the application is working on the background, it is also necessary to show a notification whenever the safety parameters are breached. This notification should be something that captivates the pedestrian from whatever they are doing on the phone.

Finally, it would be ideal if the application only did computations while the phone is being used. There are two main reasons for this: it defeats the purpose of warning a pedestrian of the danger surrounding them if they are already paying attention (the application would distract the user) and it would also consume a lot of the device's battery (this is a common problem for applications nowadays).

1.3 Thesis Structure

This thesis is separated into seven chapters, including this introductory one:

- Chapter 2 defines the current state of the environment, at which all the different technologies involved in this project are in;
- Chapter 3 states the hardware and software used to develop this application;
- Chapter 4 explains the proposed solution to achieve the planned goals in the previous section;
- Chapter 5 reports the obtained results from the proposed solution;
- Chapter 6 details the cost of the project and the social impact that the massification of this solution could have;
- Chapter 7 concludes the goals, solution and results of the project, and outlines its possible future applications.

1.4 Project Description

The goal, as mentioned before, is to send an alert of the imminent danger of a collision with a vehicle if the pedestrian is using the smartphone. If the device is not being used, no such warning should be issued. The proposed solution is an Android application running in the background when the phone is in use.

This application computes collision points by comparing the respective velocities (speed and direction) of the car and pedestrian. Once a collision is detected, the program will use a series of security parameters to determine whether a warning should be issued or not.

For this purpose, it is necessary to get the GPS locations of both the pedestrian and the car, along with their orientations. The user's location will be obtained through the Google API services and their bearing using the phone's accelerometer. The car's location and orientation will be communicated to the device through an internet connection (Wi-Fi or mobile network).

Once this information is acquired, the application will compute their trajectories and determine whether there is a collision point. If there is, it will calculate the time of arrival of each component. If both times are similar and brief, it will show a warning message to the user and send a signal to the vehicle.

For the proper testing of the application, a car with access to the internet is necessary. For this reason, it will be tested along with the iCab [11, 12], an autonomous vehicle designed by the Intelligent Systems Lab group at the Systems and Automation Engineering Department of Universidad Carlos III de Madrid. The application will also be tested for non-line-of-sight environments.

Chapter 2

State of the Art

2.1 Introduction

A diverse set of technologies and research fields have made the desired application possible. This chapter will be dedicated to listing them and explaining what they are and how they were implemented into the project. The listing is as follows:

- Autonomous Vehicles: The application attempts to improve the detection system of these vehicles, in the cases that their integrated perception systems are insufficient (like lack of visibility).
- Cars with Internet access: Essential to the project, the application will only work if the car has access to the internet in order to send its geographical information to the smartphone.
- GPS: For the respective locations of the pedestrian and the vehicle.
- P2V Communication: P2V is one of the approaches that ITS (Intelligent Transport System) uses for vehicular transportation. There are 3 methods in total: P2V (Pedestrian to Vehicle), I2V (Infrastructure to Vehicle) and V2V (Vehicle to Vehicle).

2.2 Autonomous Vehicles

An autonomous car is defined as a self-driving vehicle that has the capability to perceive the surrounding environment and navigate itself without human intervention. From here, we can clearly separate the behaviour into two parts: the perception system and navigation control [13].

The perception system of these cars may be implemented using various technologies: stereo vision, laser, radar, infrared, GPS, odometry, etc. The software will then process these inputs and obtain information about its surroundings by interpreting signs and detecting other road users (pedestrians, vehicles and cyclists).

The car will use this information to correctly navigate to its destination, following the rules of the road and avoiding collisions while making decisions on which path to take.

The autonomy of vehicles has been dreamt of since the 1920's, and have been researched until the first fully-autonomous car was developed by Carnegie Mellon University in 1984, called NavLab. The first coast-to-coast autonomous drive of the United States was in 1995, with NavLab 5 [14].

Figure 2.1: NavLab 5 CMU [15]

Since then, major development has been achieved in this technology, but very few vehicles in the market are self-driving. Most are only able to park autonomously or assist the driver. The reason for this is not only that driverless systems are still in the development stage, but the legislation, not all countries allow the circulation of these vehicles.

However, seeing how this technology is evolving, countries have begun to change their legislation [16, 17] and an increasing amount of car manufacturers are putting their vehicles autonomy in trial. Examples of this are Delphi's Audi SQ5 [18] and the autonomous Citröen Grand C4 Picasso [19].

While this section will only go into detail for the assisted driving from Ford and the autonomous vehicle from Waymo, it's worth mentioning that there are several other

models with this technology, like Volvo's pedestrian and cyclist detection system with automated brakes [20] and Mercedes-Benz Intelligent Drive within the new E-Class W213 and S213 series [21].

2.2.1 Ford's Traffic Jam Assist

Since 2015, Ford has been developing a technology that can be activated by the driver, after which the vehicle will take control of the car for a set amount of time. This system is being designed to attempt taking some of the stress and strain off of the driver, especially during heavy traffic.

Figure 2.2: Inside of a vehicle equipped with Traffic Jam Assist [22]

The system uses advanced camera and radar technology to keep the vehicle centred in the lane (lane centring assist) whilst maintaining a certain distance to the car in front (adapted cruise control Stop&Go). The front camera detects the lane markings while the radar determines the gap length that separates both vehicles [23].

2.2.2 Waymo

Google started the Google self-driving car project in 2009, which was then taken over by Waymo, an autonomous car development company belonging to the conglomerate Alphabet Inc. Since then, their vehicles have driven over 2 million miles.

The project started as a challenge to drive 100 miles fully autonomously in a Toyota Prius. The fleet was then expanded with the Lexus RX450h in 2012, the same year that the project shifted its focus from freeways and into city complexes.

Figure 2.3: Modified Toyota Prius [24]

Figure 2.4: Modified Lexus RX450h [25]

These vehicles are equipped with a laser range finder (Velodyne 64-beam laser) on the roof which generates a 3D map of the surroundings. The system then combines this 3D map with their own installed maps in order to navigate the streets while following traffic laws [26].

Their prototype vehicle, now an icon, was developed in 2014; it has custom sensors and computers, but no steering wheel or pedals. This vehicle joined the fleet in 2015. By this time, the system works using inch-precision maps that inform the car in advance of as much detail as possible (height of traffic signals, exact position of the curb, etc.) This is so the vehicle does not need to detect what its surroundings look like every time it goes through such area [27].

Figure 2.5: Waymo's self-driving car prototype [28]

2.3 Internet Access in Cars

Cars may have access to the Internet in two different ways: they may be equipped with a device with an internet connection or they can have an interface where you can connect a smartphone and access the internet through that device instead.

Vehicles with an integrated device for Internet access are called connected cars. These vehicles are equipped with a system that allows a connection to the internet through the use of telecommunication infrastructures (3G, 4G, LTE, etc). Once it secures access, it can create a Wi-Fi hotspot which allows people to connect via Wi-Fi supporting devices.

Connecting a smartphone to the car is the more popular option, as all that is necessary to implement is an interface with which to connect the device (can be Bluetooth or USB) and a smartphone application like Android Auto by Google (for Android OS) or CarPlay by Apple (for iOS).

Internet connection in cars in general has proven useful for providing functionalities and services that would otherwise not be possible. However, it is important to note that such technology can also prove to be an important distraction when driving.

2.3.1 Connected Cars

These cars have an integrated Internet connection system that revolves around the car having a SIM card slot; sometimes the car will include it with its own data plan fees, other times the owner must supply the SIM card themselves.

Automobile manufacturers such as BMW (BMW Car Hotspot or ConnectedDrive, see figure 2.6) [29], Toyota (Toyota Hotspot) [30] and Audi (Audi Connect) [31] have already launched models with such connectivity systems.

Figure 2.6: Interface for ConnectedDrive [32]

Generally, these cars will use their internet connection to provide the driver with additional services, like navigation assistance (GPS directions, traffic alerts, etc.), safety services (assistance in case of an accident or breakdown, for instance), music streaming, and any other service the manufacturer is able to design [33].

The advantage of these vehicles is they can create their own Internet Wi-Fi hotspots which other portable devices (like tablets, smartphones and laptops) can connect to and use for their own enjoyment. Similarly, using the respective application for the model, the user can use their smartphone or tablet for functions such as receiving information on the whereabouts of the vehicle and enabling or disabling the air conditioning from a distance.

The disadvantage is that they entail further expense than the smartphone interface. Not only is it expensive to implement the in-vehicle system, they also imply future charges with extra data plan fees.

2.3.2 Android Auto

This option for car Internet connectivity emerged from the fact that most of the target demographic for this system owns a smartphone. This allows for easy and malleable implementation of services and functionalities.

The main advantage of this system over the previous one is the fact that, as discussed above, the only elements necessary are a smartphone interface and a smartphone with its respective smartphone-car application.

Another reason why having a smartphone interface is favored is that it allows the driver of the vehicle to access, besides the car's own services, most of the device's features hands-free using voice controls. This way, the user can send instant messages, access their music and make phone calls without having to divert their attention from the road.

Currently, the main software applications for connecting a smartphone to a vehicle are Android Auto [34] and CarPlay [35]. To operate either of these applications, the user only needs: a smartphone with at least the minimum required updates (Android 5.0 and iOS 7.1, respectively), a car compatible with the app (there are more than 50 car brands running such system) and the proper interface (USB or Bluetooth).

Figure 2.7: Android Auto [36]

Figure 2.8: CarPlay [37]

2.4 GPS

The Global Positioning System was designed by the U. S. Department of Defense. The project was launched in 1973 and, using engineering designs from as early as the 1960's, it had its first satellite launch in 1978. The system, which became fully functional in the mid 90's, allows the operational receiver to obtain precise information about its geolocation [38].

The system can be separated into three segments: space segment, control segment and user segment.

Initially, the space segment consisted of the satellites orbiting Earth in six orbital planes, having 4 satellites on each orbit (using one as a spare). In total, there were 24 satellites orbiting at an altitude of 20 200 Km, circling the planet twice a day. Currently, there are 32 satellites in orbit in a non-uniform arrangement [39]. The reason for this is that additional satellites provide better precision and a non-uniform arrangement has been proven to be more reliable in the case that one or more satellites fail [38].

The control segment is formed by a master control station (MCS), an alternate master control station, four dedicated ground antennas and six dedicated monitor stations. The flight paths of the satellites are tracked by the monitoring stations and are then sent to the MCS. The MCS processes this information and then contacts each individual satellite with a navigational update to synchronise the constellation. This updated information is communicated to the satellites through the ground antennas [40].

The user segment is the operational receiver that processes the information received from the satellites to provide positioning, velocity and precise timing to the user. These receivers are composed of an antenna, a highly stable clock (usually a crystal oscillator), a signal conditioning circuit (generally a bandwidth filter, an amplifier and an ADC) and a processor with which to handle the information.

The user segment is, therefore, composed of hundreds of thousands secure GPS Precise Positioning Service (for military uses), and tens of millions of the Standard Positioning Service (for civil uses). The civilian usage of this technology mostly makes use of the information it can provide on position and velocity. Mainly employed for navigation, the first commercial products in the market were dedicated GPS navigation devices, which would sometimes be integrated directly into the car.

Figure 2.9: Example of a common GPS device: TomTom [41]

However, due to technology advancements, most GPS receivers are currently built into smartphones and tablets. These devices can then download and use navigation applications, both online (requires access to the internet) and offline (requires previous downloads of maps). Examples of these applications are Google Maps, MapQuest (online), HERE WeGo and Sygic (offline).

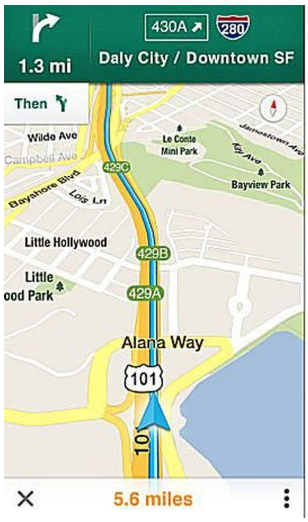

Figure 2.10: Screenshot of Google Maps application [42]

Figure 2.11: Screenshot of Sygic application [43]

2.5 P2V and V2P Communication

Intelligent Transport Systems are currently being designed to provide knowledge to vehicles in order to manage traffic and road safety. This development has yielded vehicular communications exchange of information between a vehicle and another vehicle (V2V), an infrastructure (I2V) or a pedestrian (P2V) [44].

While V2V and I2V have been developed for general road traffic support service, pedestrian-to-vehicular communication is focused on ensuring the pedestrian's safety, mainly by informing the vehicle of the position of the VRU (vulnerable road user).

The easiest way to implement this communication system is to employ devices that are already in use, such as handheld devices on pedestrians and in-vehicle systems. For pedestrians, the most logical device to use is the smartphone, as this tool is highly versatile with high computational power. In-vehicle systems are currently in development in the pursuit of more advanced warning systems (e.g., blind spot warning, intersection movement assist), and thus the incorporation of a pedestrian warning system is not far-fetched.

This technology is fairly recent, so there arent many commercialized end products using P2V communication yet. However, several companies have started to look into this development. For example, Honda's R&D department has managed to use dedicated short range communications (DSRC) technology on their prototype car to detect a pedestrian with a DSRC enabled smartphone [45].

There an application that employs P2V and V2P, along with the rest of the type of communications listed beforehand, called Comobity [46]. Comobity is an app developed by the DGT (Dirección General de Tráfico) in which the users may anonymously indicate whether they are a driver, cyclist or pedestrian and provide their location to the network. Additionally, the users may pinpoint any accident or incidence they may have found or been part of, while the network (infrastructure) may warn them back of any accident, congestion or construction site.

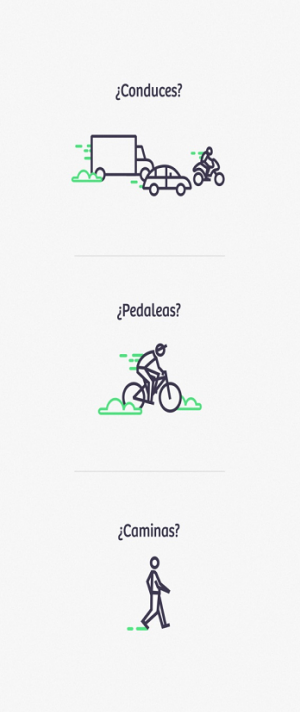

Figure 2.12: Screenshot of Comobity application [46]

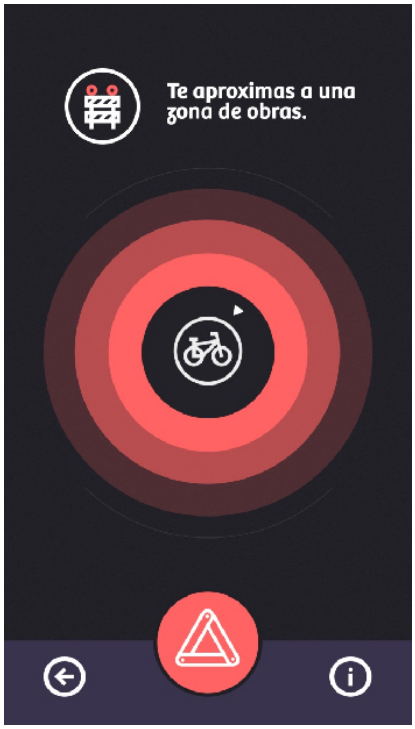

Figure 2.13: Screenshot of Commobity application [46]

2.6 Previous Mobile Warning App Versions

This particular application has been in development since 2015. So far, it obtains information from the car and calculates the collision point of the trajectories of the pedestrian and vehicle. From here, if such collision point is found, it calculates the distance from it to both road users and warns the pedestrian of the danger if a certain security threshold is breached.

However, the application will only work if the user is running it in the foreground. This is an issue, considering the low employability of this program if the user needs to be actively using it for the warning system to work.

This project is dedicated to making this application run in the background, only if the pedestrian is using the smartphone. This is due to battery consumption and the fact that it is counterproductive to have the user look at the device in order to receive a warning about their surroundings.

The application will also be designed to send a signal to the approaching car so that the vehicle can process the information in its due way (like for example, autonomously braking).

Chapter 3

Technology Framework

This chapter will describe the technology, both hardware and software, necessary to execute the project.

3.1 Hardware

Three things were used as the hardware to make this work possible: a smart device (like a phone or tablet, the target device for the application), a laptop (for the programming and simulation) and the iCab (for the simulation of other cars on the road).

3.1.1 Smart Device

The device the application was tested on was a Nexus 5X, the specifications for this phone are as follows:

Table 3.1: Mobile Phone Specifications

Operating System	Android 6.0 (upgradable to 7.1)
Dimensions	147.0 x 72.6 x 7.9 mm; 136g
Battery	2700 mAh
Rear Camera	12.3 MP, 1.55μm, f/2.0
Front Camera	5MP
Processor	Qualcomm® Snapdragon™ 808 Processor, 1.8GHz hexa-core 64-bit

Figure 3.1: Outside Appearance of the Nexus 5X [47]

The minimum requirement to run the application is to an Android operating system, with a version no older than 5.0.

3.1.2 Laptop

The laptop used for the application programming was a slightly modified Packard Bell PEW96 TK81-SB. The devices specifications are as follows:

Table 3.2: Laptop Specifications

Operating System	Windows 10 Home 64-bit
CPU	AMD Athlon™ II P340 Dual-Core Processor, 2.2GHz
Motherboard	Packard Bell EasyNote TK81 (Socket S1G4)
Graphics Card	512MB ATI AMD Mobility Radeon HD 5000
Storage	223GB (SSD)
Screen	15.6” 16:9 HD LED LCD
Optical Drive	HD-DR-ST DVDRAM GT31N
Network Adapter Type	IEEE 802.11 wireless
Connection Ports	VGA HDMI, USB 2.0 (x3)

3.1.3 iCab

The iCab project is composed of two repurposed electric golf carts (E-Z-GO D102 model), modified so that they are able to navigate autonomously. They were customised with an embedded computer with ROS (Robot Operating System) architecture and a set of sensors: a long range radar on the front, a binocular camera on the windshield and a location module on the roof.

Figure 3.2: iCab 1

Figure 3.3: iCab 2

Table 3.3: Specifications of the components in iCab

Computer	Intel Core i7, Ubuntu (x2)
Touchscreen	10” LCD
Long Range Laser	SICK LMS291, 0.25 resolution, 100m range
Binocular Camera	Bumblebee 2, 640x480 pixels, 20fps
Location Module	3DR uBlox GPS
LiDar Module	Velodyne VLP-16

3.2 Software

3.2.1 Android Operating System

Android is an open source software stack designed for touchscreen mobile devices which includes an operating system, middleware and applications. Its availability, malleability and widespread use are the reason the application will be programmed for this software.

The system is based on the Linux kernel, applications submit requests to the kernel to access the systems resources and the kernel uses the corresponding driver to execute the request.

On top of the Linux kernel, there are the middleware, libraries and API written in C programming. This software provides services beyond those available from the operating system.

On top of this layer, the application framework can be located. This layer, which provides the standard structure of the application software, includes Java-compatible libraries.

Applications, such as Contacts, Phone and Browser, will then run on said framework. This layer, programmed in Java programming language, is where the project will be developed.

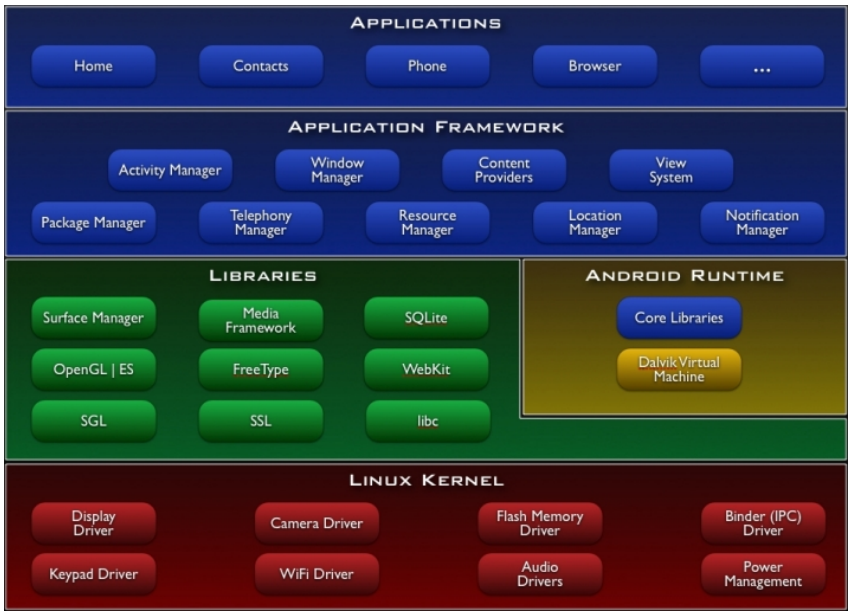

Figure 3.4: Android's architecture diagram [48]

3.2.2 Android Studio

The application was programmed using the integrated developing environment (IDE) called Android Studio. This tool, based on the IntelliJ DEA from JetBrains, is the official IDE for Android app development.

The first stable version of Android Studio was launched in December 2014, the current version used for this project is Android Studio 2.2.2. The interface and minimum system requirements of this program are shown below:

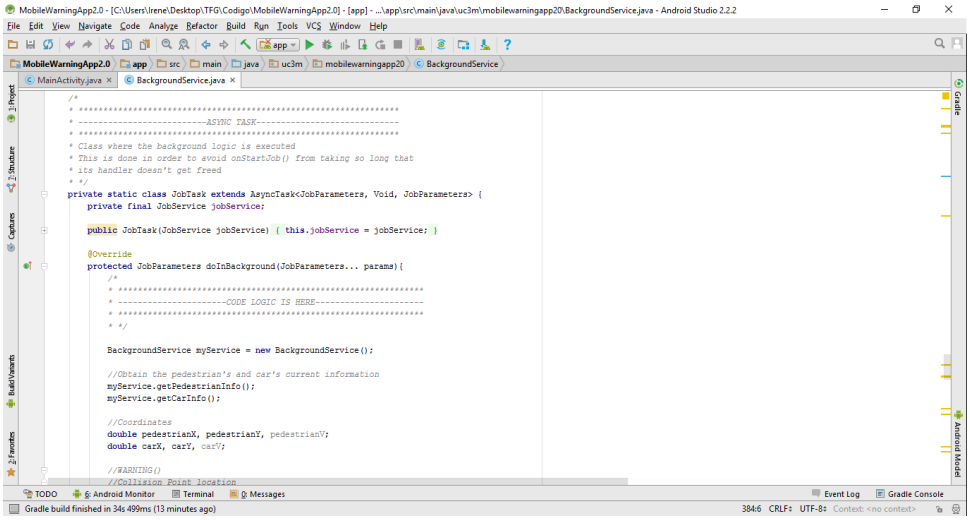

Figure 3.5: Android Studio interface

Table 3.4: Minimum system requirements for Android Studio

Operating System	Windows 7 Mac OS X 10.9.5 GNOME or KDE
RAM	3 GB (minimum); 8GB (recommended)
Disk Space	500MB (for Android Studio), 1.5GB (for Android SDK, emulators and caches)
Java Version	Java Development Kit (JDK) 8
Screen Resolution	1200x800

Chapter 4

Methodology

4.1 Introduction

The main goal of this project is to make sure the application can run in the background. The main elements of the solution will be detailed in the following sections; this section will focus on explaining the main program flow and the correlation between said elements.

Before going into the details of the implementation, it is necessary to understand the life cycle of an app and its activities and what it means to work in the background.

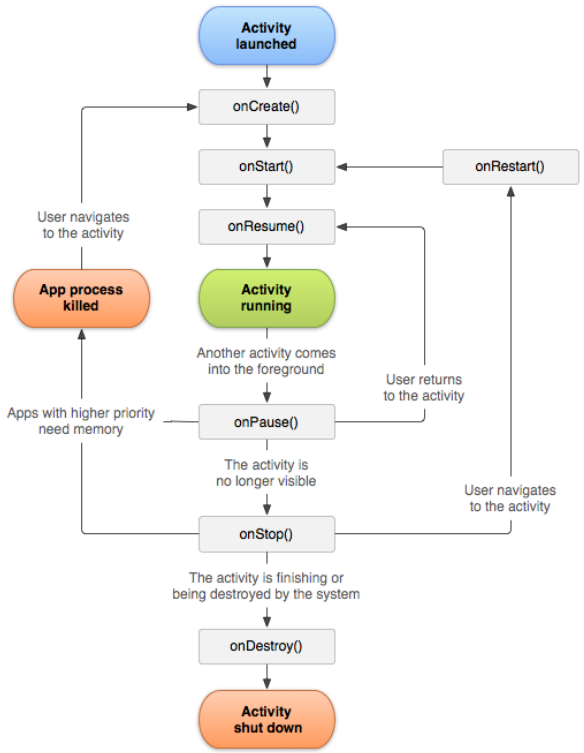

Figure 4.1: Diagram of the life cycle of an activity [49]

- onCreate(): the activity is created and being prepared to be displayed on screen
- onStart(): the activity is being displayed on the screen
- onResume(): the user can interact with the activity
- onPause(): the interaction with the activity is paused, this is because the activity has been partially covered by another one (like a pop-up window)
- onStop(): the activity is no longer being shown on screen (completely invisible to the user); this happens when the user has navigated out of the application, but the process has not been killed
- onRestart(): the activity is in the foreground again, and it needs to be prepared or display again
- onDestroy(): activity process being killed because of user shutdown or the system managing its resources

The application beforehand could only work from being created until destroyed, if the user navigated out of it, all its processes would be paused. Therefore, in order for the application to work in the background, it is necessary to implement a way to work in the cases that the application is not visible (from onStop() to onRestart()).

In the proposed solution, the background functionality is contained inside a JobService called BackgroundService. The JobService will be created when the application is created and it will be started (scheduled) when the application is stopped. It will then keep scheduling the job while the device is connected to Wi-Fi and not idle. The job will only be cancelled when the application is started and the service will be destroyed when the activity is.

When the JobService is scheduled, it starts running the method called onStartJob(). However, it is always best to offload the execution logic to another thread to avoid blocking future callbacks, especially the one meant to stop the job before it is finished, onStopJob.

For this reason, the execution code is inside an asynchronous task (AsyncTask) which implements the same code logic as in the main activity, in a more efficient manner. It calculates the position and trajectory of the pedestrian and the vehicle and obtains certain safety parameters, with which it computes whether the user should be alerted or not.

Once it decides that a warning should be issued, it does so in the form of a notification by using the NotificationManager class.

4.2 JobService

The background function is primarily done through a JobService. This is a class that extends from Service used to handle asynchronous requests that were previously scheduled.

The service is started when the application's main activity is created, but the job that it is meant to execute will only start when the activity is stopped and it is cancelled when the activity is started.

When the job is started, the function onStartJob is called. Here, the connection to the Google API Client is carried out and the listeners for the phone's accelerometer and magnetometer are declared.

As mentioned before, it is not good practice to leave the jobs execution logic in this function because doing so does not offload the main thread's handler, that can result in blocking callbacks from JobManager. For this reason, after the above actions are completed, the method creates and executes an instance of JobTask (AsyncTask).

4.2.1 Job Scheduling

The job's scheduling can be done by using the service's coded method for it (BackgroundService.scheduleJob) or an instance of JobScheduler. It was decided to use latter because this allows the cancellation of all scheduled jobs which is needed to implement in the onStart method of the main activity.

Scheduling a job requires the construction of a JobInfo object that is then passed to the JobScheduler using JobScheduler.schedule(JobInfo). The JobInfo object defines and includes the requirements for the scheduling; there must be at least one constraint. For this service, the requirements for the job to perform its execution logic is that there is a connection to a Wi-Fi network and the device is not idle.

Therefore, the job is scheduled when the application is no longer in the foreground, i.e. the main activity has called the onStop() method, as long as the requirements above are met (Wi-Fi connection and device not idle).

The JobService will then call the onStartJob() method which will start the execution thread explained in the next section. It will continue to be rescheduled until the application comes to the foreground again, i.e. the onStart() method is called. When this happens, all background jobs are cancelled since they are no longer needed.

On the next page, the flow chart of the application execution process so far is illustrated.

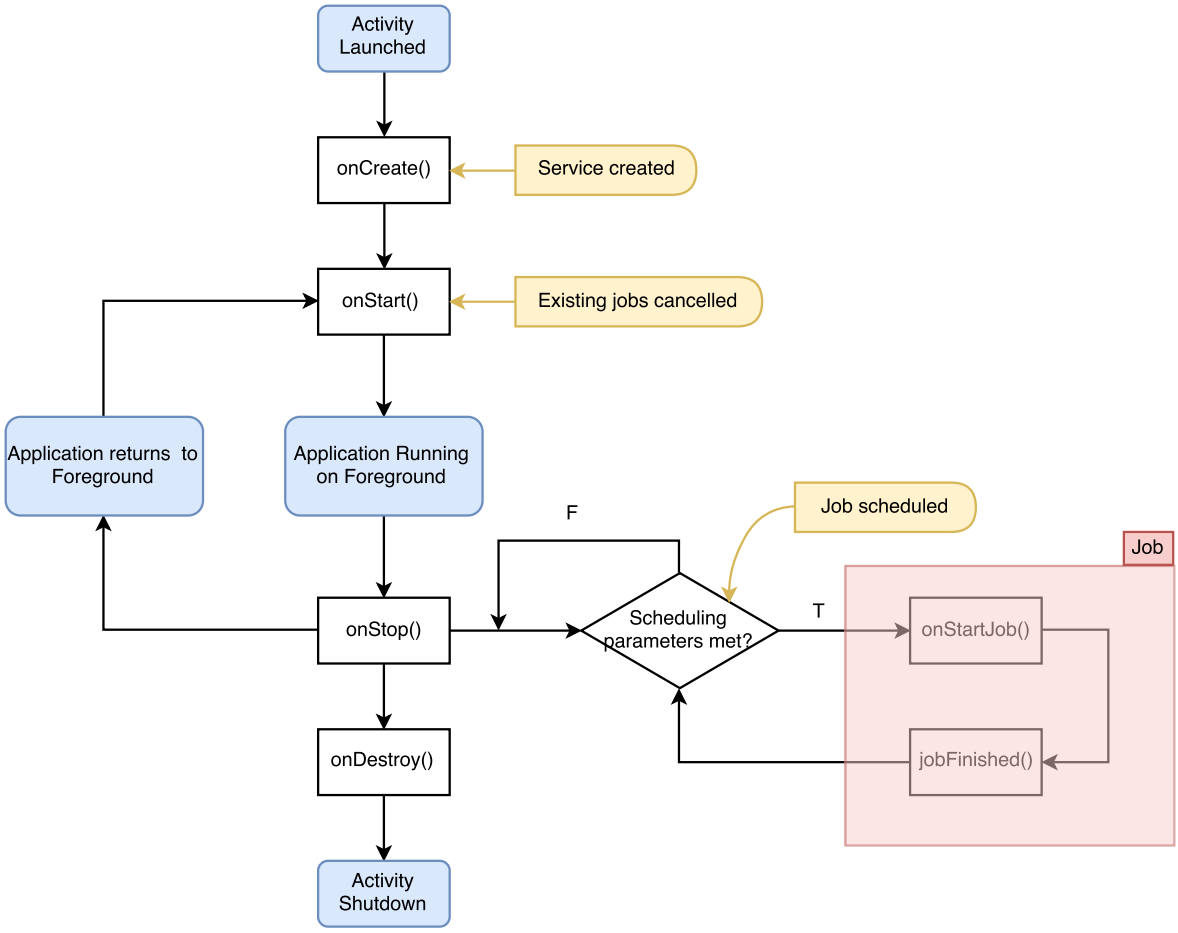

Figure 4.2: Flow chart of the Application so far

4.3 AsyncTask

This background service executes incoming jobs using a Handler running on the application's main thread. In order to avoid blocking future callbacks from the JobManager, it is necessary to transfer the execution logic to a thread or something similar. This brings into play the AsyncTask.

AsyncTask (Asynchronous Task) is important because the callbacks determine, among other things, whether a new job needs to be scheduled or, more importantly, a current job must be stopped. A job is stopped when the parameters needed for the initial scheduling are no longer being met. For example, if the job needs an internet connection and the device cannot access it anymore, the job needs to be stopped since it is ineffective to keep running it.

This class extends from Object and enables the application to perform background operations without having to manipulate threads or handlers. It is defined by 3 generic types (Params, Progress and Result) and 4 steps (onPreExecute, doInBackground, onProgressUpdate and onPostExecute).

AsyncTask must be subclassed to be used and it needs to have at least one method overwritten. In this project, the methods doInBackground and onPostExecute are overwritten.

The method doInBackground is where the execution logic of the job is located. Here, the positions of both car and pedestrian are obtained and compared, a collision point is calculated and the conclusion of whether the user must be warned is reached with the use of certain security parameters.

The method onPostExecute is called when the job is finished. It calls on the function jobService.jobFinished(JobParams, Boolean), where it uses the Boolean value true to indicate that the job should be rescheduled if the conditions of the device being connected to a Wi-Fi network and not being idle are still met.

If the conclusion of this code logic decides that the pedestrian needs to be alerted, it proceeds to use the Notification Manager, as is explained in the next section.

4.4 NotificationManager

The NotificationManager is a class used to notify the user of certain events. More specifically, the user is informed about things happening in the background. It does so by using notifications.

A notification is a message that is displayed outside of the application's activities. This message can appear in different ways: an icon on the status bar, LEDs on the device being turned on, playing a sound and vibrating, among others.

To issue a notification, it is necessary to create a NotificationManager object and use the function called notify. Said function needs a notification builder (NotificationCompat.Builder) and a notification ID (int).

The notification builder has a function called build which creates the notification that is needed to pass to the system while the notification ID is just a means for the application to identify the notification.

The following figure illustrates the execution logic of the Asynchronous Task and the use of Notifications.

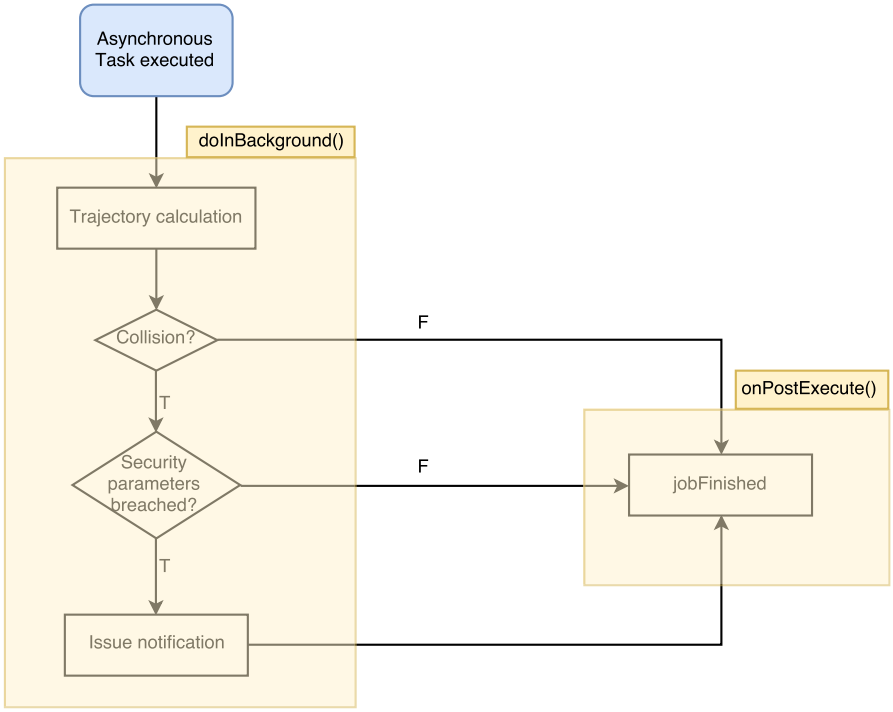

Figure 4.3: Flow chart of the Asynchronous Task, including the Notification

4.5 Concluded Remarks

This solution allows the application to successfully work in the background by scheduling the necessary computations when the proper conditions are met and alerting the user when the security parameters are breached.

In summary, the application starts a service that will remain on standby until the application is no longer in the foreground. At this point, a job is scheduled and said job offloads the execution logic onto an asynchronous task. Said task calculates the trajectories of the pedestrian user and the vehicle, computes the collision point and checks whether the pedestrian is safe or not. When it concludes that the pedestrian is in possible danger, it sends a notification to them with a warning.

This way, every goal that was declared in the introduction has been attained, except for the notification visibility. The project so far only shows a notification on the status bar of the device, but it would be ideal if said warning was shown in a more visible manner, such as a vibration or pop-up window.

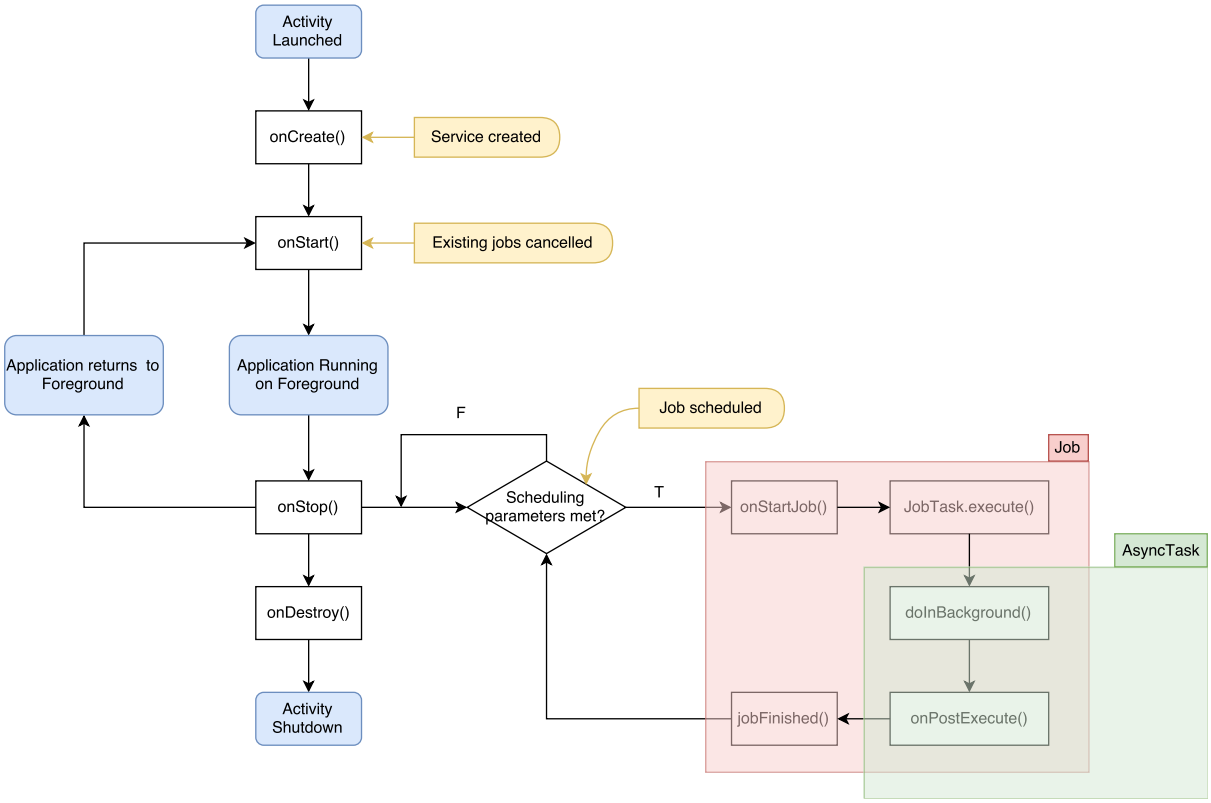

Figure 4.4: Flow chart of the whole Application

Chapter 5

Results and Discussions

5.1 Introduction

This chapter summarises the results achieved by the proposed solution, discusses the conditions required for its implementation and concludes the overall validity of this solution, considering the possible improvements that could be performed.

5.2 Objectives

The project has 3 initial goals:

- Implementation of a background functionality for the application
- Perform only background computations when device is not idle
  - Minimal resource use
  - Fewer distractions for the user
- Issue notifications when relevant

The proposed solution accomplishes all three, using JobService for the background implementation, JobScheduler in order to start the execution logic at the correct time and NotificationManager for the warnings.

5.3 Device Requirements

The libraries used in the software for the application need a minimum SDK version (also called API level) 21, 25 being the recommended one. This requires that the operating system must be at least Android 5.0 (Lollipop).

According to the Android Developers webpage, the distribution of devices running a given version of the Android platform is as follows (data collected during a 7-day period ending on May 2, 2017) [50].

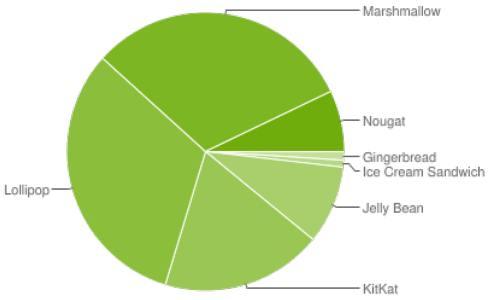

Figure 5.1: Pie chart of relative number of devices running a certain version of Android [50]

The particular numbers for the versions that can run this application (Lollipop and up) are:

Table 5.1: Distribution of devices with Android versions 5.0 or better

API Level	Android Version	Codename	Distribution (%)
21	5.0	Lollipop	8.7
22	5.1		23.3
23	6.0	Marshmallow	31.2
24	7.0	Nougat	6.6
25	7.1		0.5

This means that the application can run on approximately 70.3% of the devices. Considering that the ultimate goal of the application is to have most pedestrians using it to ensure their safety, this percentage is quite acceptable, especially when that percentage will increase over time.

5.4 Concluded Remarks

In terms of the goals of the project, as mentioned in the introduction of this chapter, all the objectives for the application development were met. The only other thing worth

mentioning is that the notifications can be improved by making them more noticeable (with a vibration, a pop-up window, etc).

Furthermore, the application can be installed for free on a smart device and can be run by 70% of current devices with an Android OS. Thus, the application has the potential to become widespread among road users, meaning better safety for all, particularly VRU (vulnerable road users).

Lastly, it is worth mentioning that the application, due to its particular conditions for background computations, uses resources responsibly. In other words, since it only performs the execution logic when the user is using the device, the application will use fewer resources than if it was always calculating trajectories. This eventually leads to a longer battery life, which has become an issue with devices recently, where most applications consume battery faster than they used to.

Chapter 6

Socio-Economic Aspect

6.1 Social Impact

This solution is made possible due to the prevalence of smart devices (phones and tablets) among people. At first glance, it seems like this is the cause of necessity of the application, but considering that pedestrians are oftentimes distracted, it is important to warn them. However, a few years ago this project would not have made any sense to develop.

This application, and others like it, may make the general public more accepting of autonomous cars. This is due to the fact that the application can also send a message to the vehicle of a pedestrian (a distracted one) is on a colliding trajectory with the car, so in cases where the system is not able to recognise the pedestrian (lack of visibility, for example), this application can warn the car as well, meaning that it can trigger the breaks of the vehicle.

Furthermore, even though technology is evolving at an extremely fast rate, it seems like it is headed towards a future where everyone is connected. This means that even if smartphones and tablets become obsolete, the idea for this project and its implementation is already developed, ready to be enabled for future devices.

6.2 Budget

6.2.1 Hardware

Table 6.1: Budget for the hardware of the application

Code	Unit	Description	Quantity	Unitary Price	Total Price
62101	Unit	**Nexus 5X** Mobile phone by the Google company, 5.2" with 1080x1920 resolution, Android OS 6.0 (upgradable) and a Qualcomm Snapdragon 808 processor	1	269.41 €	269.41 €
62102	Unit	**PEW96 TK81-SB** Packard Bell laptop unit, 15.6" with 1366x768 resolution, Windows 7 OS and Turion II 2400 MHz processor	1	53.72 €	53.72 €
62103	Unit	**Crucial® BX200** 240GB Solid State Drive storage unit with SATA 6Gb/s interface (purchase and installment)	1	95.04 €	95.04 €
62104	Unit	**iCab** Autonomous vehicle. Originally a golf cart (model E-Z-GO D102), electrically and mechanically modified to drive autonomously by using an embedded computer and sensors	1	12 396.69 €	12 396.69 €
6211		**Hardware Subtotal**	1	12 814.86 €	12 814.86 €
6212		**Hardware Total (with IVA)**	1	15 505.98 €	15 505.98 €

6.2.2 Software

Table 6.2: Budget for the software of the application

Code	Unit	Description	Quantity	Unitary Price	Total Price
62201	Unit	**Microsoft OS Windows 7** Operating System developed by Microsoft based on the Hybrid Kernel (available since 2007). To be installed on the laptop.	1	32.50 €	32.50 €
62202	Unit	**Android Studio** Integrated Development Environment for the Android platform. To be installed on the laptop.	1	0 €	0 €
62203	Unit	**Android OS 5.0** Operating System developed by Google based on the Linux Kernel (version 5.0 available since 2014). To be installed on the smartphone.	1	0 €	0 €
62204	Unit	**Ubuntu** Operating System by Canonical Ltd. and the Ubuntu community based on the Linux (Monolithic) kernel. Installed on the iCab.	1	0 €	0 €
6221		**Software Subtotal**	1	32.50 €	32.50 €
6222		**Software Total (with IVA)**	1	39.33 €	39.33 €

6.2.3 Personnel

Table 6.3: Budget for the personnel working on the application

Code	Unit	Description	Quantity	Unitary Price	Total Price
62301	Hour	**Irene Salazar Medina** Engineer in charge of developing the background functionality for the Mobile Warning application for Android and the redaction of this document.	320	7 €	2240 €
623		**Personnel**	1	2240 €	2240 €

Chapter 7

Conclusion and Future Works

First of all, this application, as explained in Chapter 5, achieves its goal of ensuring the safety of the pedestrian with the use of a device that almost every single road user has on their person. Said device issues a warning when it determines that the pedestrian is in danger.

Considering that the application can also send a warning to the car is a great advancement towards the improvement of its perception systems. Be it driven by a person or fully autonomous, GPS technology is very advantageous in several cases where the normal perception systems (driver's view or vehicle's sensors) are not enough, such as environments with low or null visibility. This asset may make autonomous vehicles not only safer, but also more accepted by the general public.

The fact that this application is very accessible due to being free and possibly installed in most Android devices is very helpful for the long term goal of widespread use. Even so, the software could always be implemented into other OS to further enhance this benefit.

In conclusion, this application has a lot of potential for mass commercialization with the safety of road users and improvement of autonomous vehicles as its goal.

7.1 Future Applications

Considering that the goal of this application is its widespread use among pedestrians, and that said application needs a P2V and V2P communication to function, it is necessary to implement said communication on a wider scale. Currently, it is necessary to input the IP address of the server (the vehicle) and the communication port between such server and the client (pedestrian). This is not feasible at larger scales, therefore maybe a more convenient way of doing this is using an infrastructure, where a specific number of servers contain information on thousands of vehicles and pedestrians, sending information on the ones closest to the pedestrian and have the application compute the required parameters out of that information.

Another point for the future of this application is to note that the pedestrian's privacy is not breached at any point, as the user's location is kept solely on their device, being the vehicle's position the one that's exchanged.

A more localised analysis of future usage of this application, using only the current iCab implementation, would be to use this system as a further help in the navigation of this prototype. Particularly if there were to be another application developed for the iCab where users from the university can 'call'the vehicle and have it take them somewhere in the campus (said users could be movement impaired, carrying heavy objects, etc.) In this case, this current application (Mobile Warning App) could have its first real implementation, since it would only need two servers (the two iCabs) and any client that would connect to them using the application.

Appendix

List of Figures

List of Tables

Appendix A

Legislation Framework

Autonomous Vehicle Testing

Due to the development of autonomous vehicles, said vehicles transcended regulation for their testing, so the Spanish legislation issued a regulation instruction to come into force on the 13th of November, 2015. This instruction updates the definition of autonomous vehicles and grants authorizations for their testing on public roads [51].

Autonomous Parking

The development and commercialization of ADAS (Advanced Driver Assistance Systems) has made the Spanish legislation consider their implementation on the roads, as such systems (like autonomous parking) have been deemed as measures that further ensure safety on the road. Thus, a regulation instruction has been issued to come into force on the 20th of January, 2016. This instruction defines assisted parking systems and their terms of use, so they can be correctly registered and allowed to be used in public roads [52].

References

[1] K. Hyers, "Global smartphone installed base forecast by operating system for 88 countries 2007 to 2017," *Strategy Analytics*, 2012.

[2] J. Poushter, "Smartphone ownership and internet usage continues to climb in emerging economies," *Pew Research Center*, 2016.

[3] W. Sigloch, "Pedestrians crossing streets: Distraction by smartphone poses risks [press release]," *DEKRA*, 2016.

[4] R. Viereckl, D. Ahlemann, A. Koster, E. Hirsh, F. Kuhnert, J. Mohs, M. Fischer, W. Gerling, K. Gnanasekaran, J. Kusber, J. Stephan, D. Crusius, H. Kerstan, T. Warnke, M. Schulte, J. Seyfferth, and E. H. Baker, "Connected car report 2016: opportunities, risk, and turmoil on the road to autonomous vehicles," *strategy&*, 2016.

[5] S. Trousdale, "Gm invests $500 million in lyft, sets out self-driving car partnership," *Reuters*, 2016.

[6] D. Primack and K. Korosec, "Gm buying self-driving tech startup for more than $1 billion," *Fortune*, 2016.

[7] C. Thompson, "Gm will use lyft to launch its first self-driving car," *Business Insider*, 2016.

[8] H. Boeriu, "Bmw sales boss talks about the future bmw inext," *BMW Blog*, 2017.

[9] B. Carson, "Travis kalanick on uber's bet on self-driving cars: 'i can't be wrong'," *Business Insider*, 2016.

[10] M. Fields, "Fords road to full autonomy," *NewCo Shift*, 2016.

[11] A. Hussein, P. Marin-Plaza, D. Martin, A. de la Escalera, and J. M. Armingol, "Autonomous off-road navigation using stereo-vision and laser-rangefinder fusion for outdoor obstacles detection," *IEEE Intelligent Vehicles Symposium (IV)*, pp. 104–109, 2016.

[12] P. Marin-Plaza, J. Beltran, A. Hussein, B. Musleh, D. Martin, A. de la Escalera, and J. M. Armingol, "Stereo vision-based local occupancy grid map for autonomous navigation in ros," *Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)*, vol. 3, pp. 703–708, 2016.

[13] K. Jo, J. Kim, D. Kim, C. Jang, and M. Sunwoo, "Development of autonomous carpart i: Distributed system architecture and development process," *IEEE Transactions on Industrial Electronics*, 2014.

[14] T. Jochem, D. Pomerleau, B. Kumar, and J. Armstrong, "Pans: a portable navigation platform," *IEEE Symposium on Intelligent vehicle*, 1995. [Online]. Available: http://www.cs.cmu.edu/~tjochem/nhaa/navlab5_details.html

[15] T. Vanderbilt, "Autonomous cars through the ages," *Wired*, 2012. [Online]. Available: https://www.wired.com/2012/02/autonomous-vehicle-history/

[16] DGT. (2015) Tráfico establece el marco para la realización de pruebas con vehículos de conducción automatizada en vías abiertas a la circulación.

[17] H. Kelly, "Self-driving cars now legal in california," *CNN*, 2012.

[18] Delphi. (2015) We've reached our destination! [Online]. Available: http://www.delphi.com/delphi-drive

[19] J. G. Casado, "El vehículo autónomo de psa finaliza el trayecto vigo-madrid," *coches.net*, 2015.

[20] autoblog, "Technology of the year: Volvo pedestrian and cyclist detection with full auto brake," *autoblog*, 2013.

[21] The new e-class. masterpiece of intelligence. Mercedez-Benz. [Online]. Available: https://www.mercedes-benz.com/en/mercedes-benz/vehicles/passenger-cars/e-class/the-new-e-class-the-most-intelligent-business-saloon/

[22] M. Karkafiris, "Ford developing semi-autonomous traffic jam assist and remote parking systems," *CarScoops*, 2015. [Online]. Available: http://www.carscoops.com/2015/12/ford-developing-semi-autonomous-traffic.html

[23] (2015) New ford autonomous tech turns traffic jams into chill time and parks your car by remote control. Ford. [Online]. Available: http://www.ford.co.uk/experience-ford/AboutFord/News/VehicleNews/2015/New-ford-autonomous-tech

[24] J. Holmes, "Tesla may work with goodle on autonomous cars," *Motor Trend*, 2013.

[25] A. Iliaifar, "Google adds lexus rx450h to ongoing self-driving car project," *Digital Trends*, 2012. [Online]. Available: https://www.digitaltrends.com/cars/google-adds-lexus-rx450h-to-ongoing-self-driving-car-project/

[26] E. Guizzo, "How google's self-driving car works," *IEEE Spectrum*, 2011.

[27] A. C. Medrigal, "The trick that makes google's self-driving cars work," *The Atlantic*, 2014.

[28] Waymo. [Online]. Available: https://waymo.com/journey/

[29] BMW Connected Drive. BMW. [Online]. Available: http://www.bmw.com/com/en/insights/technology/connecteddrive/2013/

[30] Toyota hotspot. Toyota. [Online]. Available: https://www.toyota-europe.com/service-and-accessories/genuine-accessories/hotspot

[31] Audi connect infotainment. Audi. [Online]. Available: https://www.audi.co.uk/audi-innovation/advanced-technologies/audi-connect.html

[32] (2014) Bmw connecteddrive review. [Online]. Available: http://www.trustedreviews.com/bmw-connecteddrive-review

[33] everis, "everis connected car report," 2014. [Online]. Available: http://s3-eu-west-1.amazonaws.com/e17r5k-datap1/everis_documents_downloads/everis+connected+car+report.pdf

[34] One app, two ways to use android auto. Android. [Online]. Available: https://www.android.com/auto/

[35] Apple carplay. the ultimate copilot. Apple. [Online]. Available: https://www.apple.com/ios/carplay/

[36] P. Nickinson, "This is what it's like to use android auto on a car head unit," *Android Central*, 2016. [Online]. Available: https://www.androidcentral.com/take-ride-us-and-android-auto-2-pm-et

[37] R. Amadeo, "One week with apple's carplay," *Ars Technica*, 2016. [Online]. Available: https://arstechnica.com/apple/2016/01/one-week-with-apples-carplay/

[38] (2017) United States Library of Congress. [Online]. Available: http://www.loc.gov/rr/scitech/mysteries/global.html

[39] Current gps constellation. United States Naval Observatory. [Online]. Available: http://tycho.usno.navy.mil/gpscurr.html

[40] Usno navstar global positioning system. United States Naval Observatory. [Online]. Available: http://tycho.usno.navy.mil/gpsinfo.html

[41] T. John, "Tomtom via series gps devices in india: Hands-on review," *Techulator*, 2012. [Online]. Available: http://www.techulator.com/resources/7936-TomTom-Via-125-120-100-Series-GPS-Hands-on-review.aspx

[42] F. Zahradnik, "Review: Google maps for iphone - maps and gps navigation," *Lifewire*, 2016. [Online]. Available: https://www.lifewire.com/google-maps-for-iphone-1683478

[43] T. Dawson, "Sponsored app review: Sygic: Gps navigation & maps," *Android Headlines*, 2013. [Online]. Available: https://www.androidheadlines.com/2013/12/featured-app-review-sygic-gps-navigation-maps.html

[44] M. Nagai, K. Nakaoka, and Y. Doi, "Pedestrian-to-vehicle communication access method and field test results," *2012 International Symposium on Antennas and Propagation (ISAP)*, 2012.

[45] R. Eagles, S. Kinkade, C. Naughton, and S. Pines, "Honda demonstrates advanced vehicle-to-pedestrian and vehicle-to-motorcycle safety technologies," *Honda News*, 2013.

[46] (2015) Comobity: La app que conecta y protege a los conductores, ciclistas y peatones. DGT. [Online]. Available: http://www.dgt.es/es/prensa/notas-de-prensa/2015/20151112-App-Comobity-DGT.shtml

[47] Google. [Online]. Available: https://www.google.com/intl/es_es/nexus/5x/

[48] Android architecture. eLinux. [Online]. Available: http://elinux.org/Android_Architecture

[49] The activity lifecycle. Android Developers. [Online]. Available: https://developer.android.com/guide/components/activities/activity-lifecycle.html

[50] (2017) Platform versions. Android Developers. [Online]. Available: https://developer.android.com/about/dashboards/index.html#Platform

[51] *INSTRUCCIÓN 15/V-113*, Ministerio del Interior Std., 2015. [Online]. Available: http://www.dgt.es/Galerias/seguridad-vial/normativa-legislacion/otras-normas/modificaciones/15.V-113-Vehiculos-Conduccion-automatizada.pdf

[52] *INSTRUCCIÓN 16 TV/89*, Ministerio del Interior Std., 2016. [Online]. Available: http://www.dgt.es/Galerias/seguridad-vial/normativa-legislacion/otras-normas/modificaciones/2016/Instruccion_16_TV_89_Estacionamiento_asistido_vehiculos_motor.pdf